2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Fast Train: A Computationally Efficient Train Routing and Scheduling Engine for General Rail Networks

Lingyun Meng, Xuesong Zhou

Abstract—**A number of tactical and operational applications, such as train timetabling and dispatching, require sophisticated and computationally efficient software to optimize train schedules. An open-source train scheduling package, namely FastTrain has been designed and implemented to generate feasible schedules and optimality gap estimates for trains on a generic rail network with both single and double tracks. This paper describes its two major modelling components: (1) a network cumulative flow model to capture the complex safety rules of infrastructure resources, and (2) a Lagrangian relaxation solution framework which efficiently decomposes the original multi-trains complex model to a set of single train oriented subproblems. Two experimental cases based on the adapted datasets released by INFORMS RAS are conducted to demonstrate the effectiveness and efficiency of the developed algorithm under different network and data availability conditions.**

Index Terms— **Train Routing, Train Scheduling, Resource Constraints, Lagrangian Relaxation, Shortest Path Algorithm**

I. Introduction

Providing punctual and reliable services is one of the main goals in railway management and operations. As tactical plans, train timetables are programmed and updated every year or every season (offline) to define traveling routes in the rail network and detail arrival/departure times of trains. In daily (online) train operations, various types of perturbations may disturb scheduled train running times, dwelling and departing events, thus causing primary delays to the planned train schedule. Due to the high interdependency between trains, primary delays could propagate as knock-on delays to other trains on a network.

Most existing computer-aided systems for Train Routing and Scheduling (TRS) focus on information management, rather than providing decision supports to planners or dispatchers. Both offline tactical planning and online operational management call for a theoretically rigorous and computationally efficient methodology to automatically generate schedules which consist of routes and arrival/departure times for trains on a general rail network.

FastTrain, an open-source mesoscopic train scheduling package implemented in C++, in conjunction with the related mathematical programming model implemented in CPLEX, has been developed to provide a rail network scheduling tool to train timetable planners, train dispatchers, and researchers. This train scheduling software can be downloaded from http://code.google.com/p/fast-train/ and the mathematical model can be found at http://code.google.com/p/networked-train-scheduling/. In general, FastTrain aims to:

(i) provide an open-source software prototype to support transportation researchers and software developers to expand its range of capabilities to various rail traffic planning and management applications;

(ii) serve as a decision-support software core engine to provide rail planners and dispatchers with candidate train schedules;

(iii) provide a free, educational tool for students to understand the complex decision-making process in rail transportation planning and optimization processes.

This paper is organized as follows. Section II introduces the overall model design of FastTrain. This is followed by key modelling techniques in section III and a Lagrangian relaxation based solution framework in section IV. The paper closes with section V demonstrating numerical experiments using adapted datasets released by INFORMS RAS.

II. Model Design and System Structure

The software architecture of FastTrain is designed to combine many rich modeling capabilities into an open-source train scheduling model. By designing FastTrain in a modularized fashion, the software can also meet future needs from rail operations researchers and software developers to continue to expand its capabilities in, for example, the line planning problem by introducing passenger flows into the train scheduling model. The overall software structure, illustrated in Fig. 1, integrates six major modeling components:

(1) Space-time network generation. Based on the input files which include topology of the physical infrastructure (nodes, links) network, as well as train attributes (candidate links the train may use), this module builds up a space-time network. This extended space-time network is used by the time-dependent shortest algorithm in module (2).

(2) Time-dependent shortest (TSP) path finding. The space-time network is first combined with resource prices updated in module (5), and then the TSP algorithm searches for the least-generalized-cost path from train's origin node to destination node which consists of the route and arrival/departure times. The TSP algorithm is used in modules (4) and (5).

*We acknowledge the support by the State Key Laboratory of Rail Traffic Control and Safety (contract No RCS2013ZZ001), Beijing Jiaotong University, and the Fundamental Research Funds for the Central Universities of China (contract No 2014JBZ008, 2013JBM146).

L. Meng is with State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, ShangYuanCun 3#, HaiDian District, 100044, Beijing, China. lymeng@bjtu.edu.cn

X. Zhou is with School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85287, Arizona, USA. xzhou74@asu.edu

978-1-4799-6078-1/14/$31.00 ©2014 IEEE

(3) Train generation, which combines basic information (origin, destination, speed multiplier) with the additional time-of-day departure time profile to generate trains. Generated trains are used in modules (4) and (5).
(4) Priority rule-based algorithm, which first generates a rank list of trains according to their Lagrangian multipliers updated in module (5), and then apply the TSP algorithm to route and schedule train in a train-by-train fashion.
(5) Lagrangian relaxation algorithm, which is the core of the solution framework. Its main function is to update resource prices (i.e., Lagrangian multipliers) according to resource utilization conditions in the previous iteration, build up space-time networks for trains, find the least cost paths and compute the lower bound by the time-dependent shortest path algorithm, call the priority rule-based algorithm to compute upper bound, and output computational results at the current iteration.
(6) Parallel computing technique. In order to fully utilize the efficient and easy decomposition mechanism enabled by the Lagrangian relaxation solution framework, FastTrain is designed to have potential interfaces for using parallel solution techniques to speed up the solution process in the near future.

The major focus of this paper is on describing: (a) the key modelling techniques in the Lagrangian relaxation solution framework, (b) the Lagrangian relaxation solution algorithm, (c) the time-dependent shortest path algorithm, and (d) the priority rule-based feasible schedule generation algorithm.

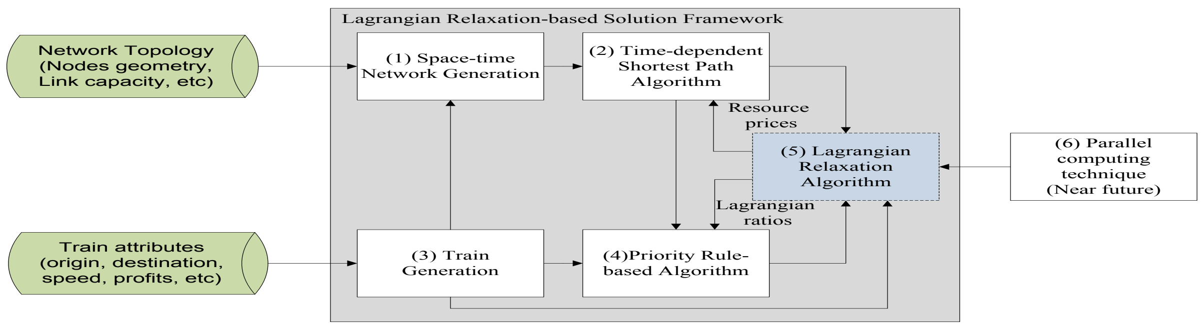

Fig. 1 Software system architecture with key modeling component

III. KEY MODELING TECHNIQUES

A. Space-time representation of physical rail network

In FastTrain, the input physical network is first transformed into a space-time network by extending each physical node to a number of vertexes according to discretized time units and constructing corresponding arcs. For illustrative purposes, Fig. 2 depicts a transformation of the left physical network (4 nodes, 5 links) to the right space-time network.

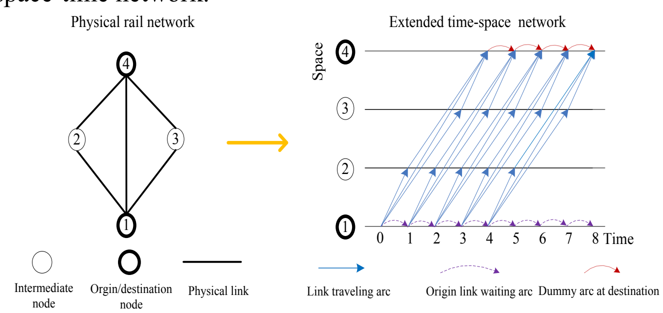

Fig. 2 Space-time network representation of physical network

Based on the extended space-time network, a set of inflow decision variables $\mu_f(i,j,t)$ and $\mu'_f(i,j,t')$ are introduced to represent the route selection and corresponding arrival/departure times. $\mu_f(i,j,t)=1$ represents train f arriving at the upstream node i of link (i,j) at time t, $\mu_f(i,j,t)=0$, otherwise; $\mu'_f(i,j,t')=1$ represents train f departing from the downstream node j of link (i,j) at time t', $\mu'_f(i,j,t')=0$, otherwise. $\mu_f(i,j,t)=1$ and $\mu'_f(i,j,t')=1$ means that the arc from vertex (i,t) to (j,t') is selected for train f.

B. Transformation of inflow variables to cumulative flow variables

The TRS problem needs to precisely model spatial and temporal occupancy of trains on infrastructure with respect to various safety headway constraints. The above inflow variables-based modeling method can nicely formulate temporal and spatial occupancy of trains on the network, but modeling safety headways between adjacent trains is still challenging.

Cordeau et al. (1998), Hansen (2010), and Pacciarelli (2013) reviewed many key modeling aspects of rail operations. The so-called "big-*M*" based method (e.g. used in Zhou and Zhong, 2007) has been widely used to represent either-or type of train ordering constraints by introducing a sufficiently large positive number *M*. Considering the complexity of TRS models, many studies are devoted to efficient decomposition mechanisms for reducing the model complexity, and heuristic algorithms for obtaining feasible solutions within a reasonable computational time (e.g., train-based decomposition by Carey and Lockwood, 1995, Lee and Chen, 2009). However, in TRS models under the context of general rail networks, the capacity constraints are extremely difficult to decompose into different solution branches, as there are a large number of possible routes leading to very different resource usage combinations. A recent study on advanced branching methods under stochastic capacity conditions can be found in Meng and Zhou (2011). By seamlessly linking the inflow variables with cumulative flow variables, FastTrain aims to easily model both temporal and spatial occupancy of trains on tracks as well as safety time headways under a multi-track context. Moreover, the track capacities in a network can be modelled through side constraints, which can be dualized in a Lagrangian relaxation framework. Thus it can provide a tractable foundation for efficient train-based decomposition.

We now introduce a set of cumulative flow based decision variables, $a_f(i,j,t)$, $d_f(i,j,t)$ which are 0-1 binary variables. $a_f(i,j,t)=1$ if train *f* has already arrived at link (i,j) by time *t*, and otherwise $a_f(i,j,t)=0$. $d_f(i,j,t)=1$, if train *f* has already departed from link (i,j) by time *t*, and otherwise $d_f(i,j,t)=0$.

Fig. 3 illustrates how to use cumulative arrival/departure variables to describe the link selection and arrival/departure times for train *f* at link (1, 2).

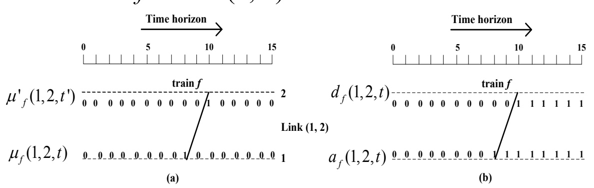

Fig. 3 Transformation of inflow variables to cumulative flow variables

As shown in Fig. 3(a), train *f* arrives at link $(1,2)$ at time 8 and departs at time 10, with $\mu_f(1,2,8)=1$ and $\mu'_f(1,2,10)=1$. With cumulative flow variables, as shown in Fig. 3(b), we have: $a_f(1,2,t)=0$ for $t<8$ and $a_f(1,2,t)=1$ for $t\geq 8$; $d_f(1,2,t)=0$ for $t<10$ and $d_f(1,2,t)=1$ for $t\geq 10$. The relationship between inflow variables and cumulative flow variables can be represented by Eqs. (1) and (2).

$$\mu_f(i,j,t)=a_f(i,j,t)-a_f(i,j,t-1) \quad (1)$$

$$\mu'_f(i,j,t)=d_f(i,j,t)-d_f(i,j,t-1) \quad (2)$$

Furthermore, $a_f(i,j,T)=1$ can demonstrate that the link (i,j) is used by train *f* to traverse the network, where *T* means the planning horizon.

C. Modeling safety headway by cumulative flow variables

Before detailing how to use cumulative flow variables to model safety headway, we first introduce modeling assumptions and partial key notations used in the model. For full notation and mathematical formulation of the TRS model inside FastTrain, we refer to a recent paper by Meng and Zhou (2014).

FastTrain has the following assumptions:

(1) A train is modeled as a virtual object without physical length for simplicity.

(2) The segment between stations is modelled as a series of block sections for a unidirectional double-track rail line and as one block section for a bidirectional single-track line. A block section is denoted as a link in our paper.

(3) A station can be represented as a single node in an aggregated manner, a single link corresponding to a main track, or a subnetwork with a main track (i.e. non-siding track) and a number of siding tracks being modelled as a set of links.

(4) A node can be viewed as a relevant point of network. It can be used to model a station in an aggregated manner, a beginning/ending point of a block section or a main/siding track in station, or a point of merging/diverging of tracks.

(5) Only one train is permitted on a link at any given time.

(6) The granularity of time is one minute.

Based on the set of cumulative arrival and departure variables, a set of shifted cumulative flow variables $a_f(i,j,t+g)$ and $d_f(i,j,t-h)$ is introduced to represent safety time headway *g* and *h*. We can represent the spatial occupancy of a train through a simple equation $y_f(i,j,t)=a_f(i,j,t+g)-d_f(i,j,t-h)$, where $y_f(i,j,t)$ is a set of 0-1 binary occupancy variables with $y_f(i,j,t)=1$, if train *f* occupies link (i,j) at time *t*, and otherwise $y_f(i,j,t)=0$. For example, let's assume $g=h=1$, the gray rectangle block in Fig. 4 corresponds to $y_f(i,j,t)=1$ for $t=7\ldots10$, $y_f(i,j,t)=0$ otherwise, which means train *f* occupies link (i,j) from time 7min to time 10min.

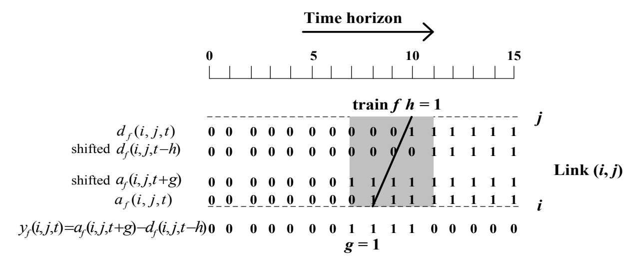

Fig. 4 Spatial occupancy of link (i,j) by train f between time 7 min and 11 min.

We now move to a double-track segment case with four sequencing links e_1, e_2, e_3 and e_4. As shown in Fig. 5, the stacks of gray rectangles represent the detailed occupancy by trains *f* and *f* ' in a space-time network. On a double-track segment, it is possible that two trains are running at the same time (e.g. at minute 14 in Fig. 5), while there is only one train being allowed on each link. The proposed formulation using cumulative flow variables can nicely capture the above two requirements through $y_f(i,j,t)+y_{f'}(i,j,t)\leq 1$.

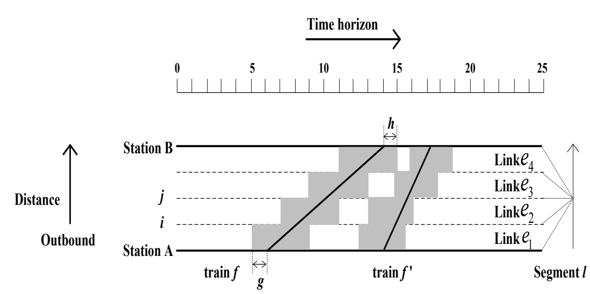

Fig. 5 Link decomposition of a double-track segment l between stations A and B.

A single-track case is illustrated in Fig. 6. We need to introduce directed link e from station i to j and link e' from station j to i, in order to allow trains running on opposite directions. Let us consider train f using e and train f' using e'. Since links e and e' correspond to the same segment, we can use a constraint of $y_f(i,j,t) + y_{f'}(j,i,t) \le 1$ to model the safety headway requirement.

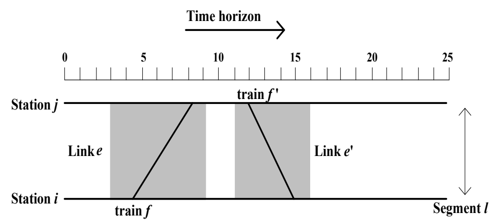

Fig. 6 Two links corresponding to a single-track segment l from station i to j.

Based on variables $a_f(i,j,t)$, $d_f(i,j,t)$, $y_f(i,j,t)$, safety headway g and h, and $Cap(i,j,t)$, which is the (input) capacity of link (i, j) at time t, the key safety headway constraints can be easily modeled by constraints (3) and (4), where E_f indicates the set of links train f may use.

Link occupancy indication constraints:

$$y_f(i,j,t) = a_f(i,j,t+g) - d_f(i,j,t-h), \forall f, (i,j) \in E_f, t \quad (3)$$

Link capacity constraints:

$$\sum_{f:(i,j)\in E_f} y_f(i,j,t) + \sum_{f':(j,i)\in E_{f'}} y_{f'}(j,i,t) \le Cap(i,j,t), \ \forall i,j,t \quad (4)$$

Constraints (3) map $y_f(i,j,t)$ with $a_f(i,j,t+g)$ and $d_f(i,j,t-h)$. The first term is 1 if train f has started occupying link (i,j) by time t and the second term will be 1 if train f has ended occupying link (i,j) by time t. Therefore, the only trains that contribute a value of 1 to the difference $a_f(i,j,t+g) - d_f(i,j,t-h)$ represent the trains that are occupying link (i,j) at time t, i.e., $y_f(i,j,t) = 1$.

Furthermore, constraints (4) make sure the number of trains that are occupying link (i,j) are less that the capacity of link (i,j), which implicitly ensures safety time headways between trains.

The additive structure of capacity usage (left portion of constraints (4)) forms the foundation of the reformulated model, as it can decouple the original problem into many train-specific subproblems. The mechanism is later used by a Lagrangian relaxation solution framework in section IV.

It should be noted that the cumulative flow decision variables enable a simultaneous train routing and scheduling solution approach. This approach implicitly enumerates all possible routes in the network and jointly optimizes train routes and arrival/departure times. Shown by Meng and Zhou (2014), the proposed simultaneous approach is capable of obtaining better solutions compared widely used traditional sequential train routing and scheduling approach.

IV. LAGRANGIAN RELAXATION FRAMEWORK

To provide useful optimization quality measures for solutions of the TRS problem, we use a Lagrangian relaxation solution framework to: (i) solve the dualized problem for constructing a tight lower bound and (ii) provide a good base solution for generating feasible solutions with valid upper bounds.

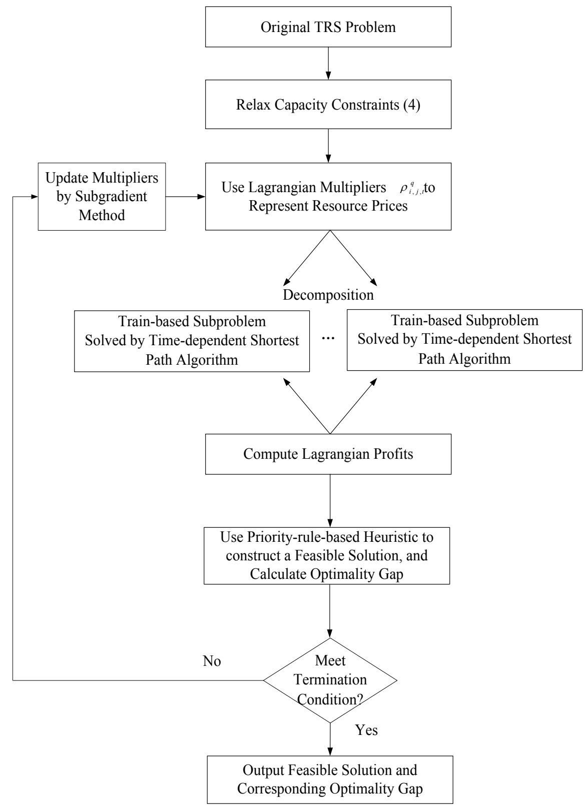

Fig. 7 Process of the proposed Lagrangian relaxation solution framework

The Lagrangian relaxation solution framework is depicted in Fig. 7. In this framework, capacity constraints (4) are considered as hard constraints and relaxed as a penalty term in the objective function, i.e., the second portion of Eq. (5). The first portion is the objective function in the original model which aims to minimize total deviation time of all involved trains, where $D(s_f)$ indicates train f' preferred arrival time at its destination node s_f. If one aims to minimize total completion time of all trains, he/she can easily implement this objective by setting $D(s_f)=0$ for all trains.

$$\min Z = \sum_f \left\{ \left| \sum_t t \times \sum_{i:(i,s_f)\in E^s(s_f)\cap E_f} \left[d_f(i,s_f,t) - d_f(i,s_f,t-1) \right] - D(s_f) \right| \right\} + \sum_{i,j}\sum_t \rho_{i,j,t} \times \left\{ \sum_{f:(i,j)\in E_f} y_f(i,j,t) + \sum_{f':(j,i)\in E_{f'}} y_{f'}(j,i,t) - Cap(i,j,t) \right\} \quad (5)$$

In Eq. (5), we define a set of Lagrangian multipliers denoted as $\rho^q_{i,j,t}$, where q means the iteration number and then decompose the original TRS problem to a set of train-based subproblems LR_f as in Eqs. (6) and (7).

$$\max_{\rho_{i,j,t}\ge 0} LR = -\sum_{i,j}\sum_t [\rho_{i,j,t} \times Cap(i,j,t)] + \min \sum_f LR_f \quad (6)$$

where

$$LR_f = \left|\sum_t t \times \sum_{i:(i,s_f)\in E^S(s_f)\cap E_f} [d_f(i,e_f,t) - d_f(i,e_f,t-1)] - D(s_f)\right| + \sum_{(i,j)\in E_f} \sum_t [\rho_{i,j,t} \times y_f(i,j,t)] \tag{7}$$

In a subproblem with train *f*, the objective is to find the time-dependent least (generalized) cost path of train *f* from its origin node to its destination node. The generalized cost includes two parts, one is the schedule cost and the other is resource cost. For an offline train timetabling problem, schedule cost refers to total travel time of train *f* from its origin node to destination node, while for an online train dispatching problem, schedule cost is expressed as the deviation time of train *f* at its destination node, compared to its preferred arrival time. The resource cost of train *f* for traversing a network from the origin node to the destination node is computed by summing $\rho_{i,j,t}$ over all selected links within associated time spans.

We use a label-correcting based time-dependent shortest path algorithm to solve each subproblem. The framework for the shortest path algorithm is adapted from the algorithm by Ziliaskopoulos and Mahmassani (1993) for highway or urban road networks, where the scan eligible (SE) list only contains spatial nodes rather than time-space vertexes. Another special feature of our proposed algorithmic framework is that we allow trains waiting at cells so there are multiple possible cell running times from the same time. In contrast, the deterministic algorithm for highway networks typically considers one instance of time-dependent arc travel time. Interested readers can also refer to Pallottino and Scutellà (1998) for more details on time-space network construction and other time-dependent shortest path algorithms using different mechanisms of constructing and updating SE lists.

After solving the train-based subproblems, we compute Lagrangian profits for each train and then rank the trains by decreasing values of Lagrangian profits. The Lagrangian profit of each train is the ratio of total free-flow travel time divided by total travel time in the dual solution.

A priority rule-based heuristic algorithm is then used to transform dual solutions to feasible solutions. The priority of trains is determined by corresponding Lagrangian profits.

Based on the dual solutions and feasible solutions, we compute the optimality gap at current iteration and then check whether the termination condition is met. We use termination criteria as: if $q > Q_{\max}$ (a predetermined maximum number of iteration) or the gap is less than the predetermined value, or the quality of feasible solution does not improve for a certain number of iterations (e.g. 100), then the algorithm ends.

If the termination condition is met, then the algorithm outputs feasible solutions with quality measures (i.e. optimality gap). Otherwise, a subgradient method is invoked to update Lagrangian multipliers and then move to the next iteration.

V. Numerical Experiments

We adapt a network (as shown in Fig. 8) from the INFORMS RAS problem competition in 2012 (INFORMS RAS, 2012) as our main test dataset. It consists of 78 nodes and 88 links, with a total track length of 151.4 km. MOW in Fig. 8 represents Maintenance of Way where the corresponding cells are unavailable to rail traffic due to repair or inspection activities.

In a 24-hour planning horizon, 25 trains need to be scheduled. The initial delays of trains occur at the trains' origin node and follow the uniform distribution with a range between 10 and 20 minutes.

The proposed Lagrangian relaxation solution framework in FastTrain is implemented as a customized algorithm by Visual C++ 2008 on a Windows 7 X64 professional platform. All the experiments are performed on a Lenovo ThinkPad T530 laptop with 2.9 GHz Intel i7 CPU and 4 GB memory.

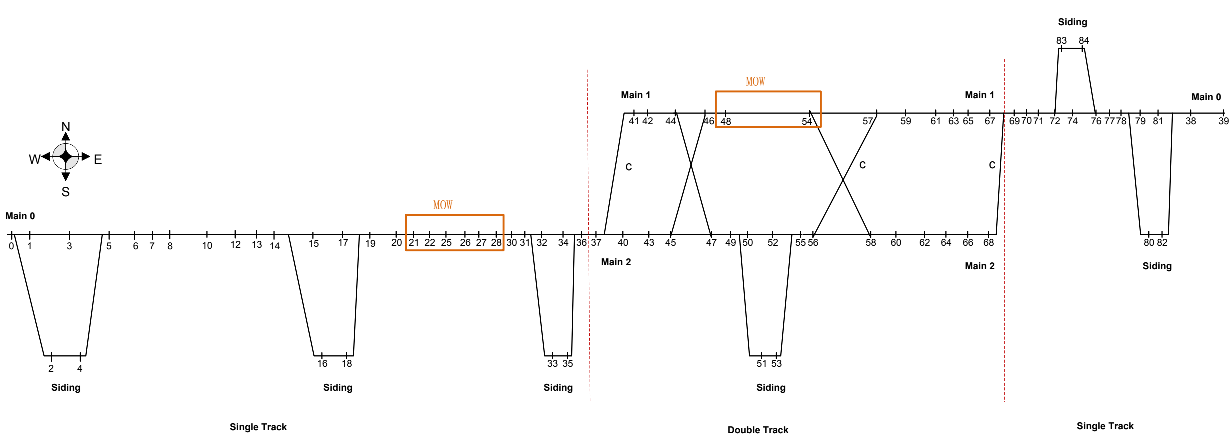

Fig. 8 An experimental rail network adapted from INFORMS RAS (2012)

Recall that the model in FastTrain allows for a simultaneous train routing and scheduling approach fully considering all possible routes on the network, while the traditional sequential rerouting and rescheduling approaches compute schedules under specifying a number of routes K for each train f, where $K=\left|P_f\right|$. Note that P_f contains a number of routes sorted in ascent order according to route length. For instance, K=1 means that there is one route (i.e., the shortest one) from origin to destination.

We next aim to demonstrate the comparative benefits of the proposed model compared to sequential train routing and scheduling approaches with test cases where K=1, 3, 5 and 10 respectively.

Fig. 9 and Fig. 10 examine the upper bound and optimality gap, respectively, under different model settings. Compared to K=1, with increasing K for each train, the upper bound solution quality generally improves corresponding to K=3, 5, and 10, although the quality with K=5 is a little worse than that of K=10 due to limited iterations. We compare the case with K=5 and K=10 with the proposed model that considers all possible available routes. Interestingly, one can still observe additional benefits in terms of 12.6% and 10.7% optimality gap reductions. This result indicates that the proposed model offers an ideal solution benchmark for evaluating existing solution strategies, typically obtainable by a sequential routing-scheduling process based on relatively limited available route sets.

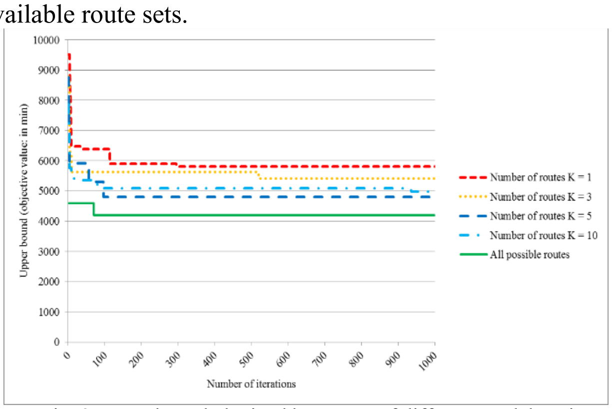

Fig. 9 Upper bound obtained by means of different model settings.

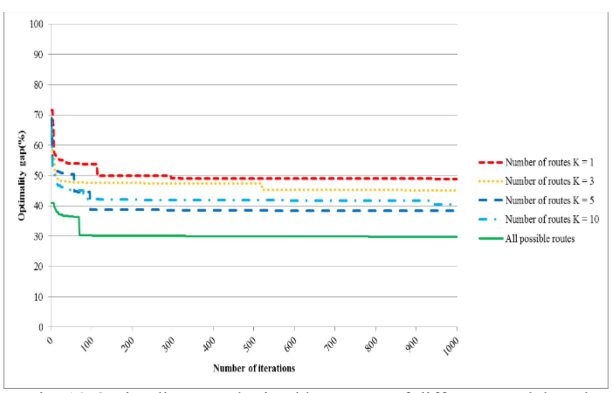

Fig. 10 Optimality gap obtained by means of different model settings.

In addition, thanks to the efficient implementation of the proposed Lagrangian relaxation solution framework implementation, the proposed model can find a good upper bound quickly within less than 100 iterations, which takes less than 1 minute of CPU time. Moreover, the lower bound also quickly becomes stable within 3 minutes, which leads to an optimality gap indictor of the obtained upper bound solution.

VI. Conclusions

In this paper, a cumulative flow based train routing and scheduling model is proposed to allow simultaneously routing and scheduling trains on a general rail network. Based on the proposed model, an open-source software package called FastTrain was developed to rapidly generate train schedules with useful quality measures.

Our future research will develop different primal-dual optimization methods, e.g. branch and bound and branch and price, to iteratively generate feasible solutions with smaller optimality gaps. We are currently also in the process of implementing parallel computing techniques to speed up the solution process for shortest path subproblems (for each train) which can be easily decomposed to different CPU cores.

Acknowledgment

The work of the first author was jointly supported by the State Key Laboratory of Rail Traffic Control and Safety (contract No RCS2013ZZ001), Beijing Jiaotong University and the Fundamental Research Funds for the Central Universities of China (contract No 2014JBZ008, 2013JBM146).

References

[1] Carey, M., Lockwood, D., 1995. A model, algorithms and strategy for train pathing. Journal of the Operational Research Society 46 (8), 988-1005.

[2] Cordeau J., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. Transportation Science 32 (4), 380-404.

[3] Hansen, I. A., 2010. Rail Network timetabling and dynamic traffic management. International Journal of Civil Engineering 8 (1), 19-32.

[4] INFORMS RAS, 2012. Problem description and released data set for problem solving competition- 2012. http://www.informs.org/Community/RAS/Problem-Solving-Competition/2012-RAS-Problem-Solving-Competition.

[5] Lee, Y., Chen, C., 2009. A heuristic for the train pathing and timetabling problem. Transportation Research Part B 43 (8-9), 837-851.

[6] Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. Transportation Research Part B 45 (7), 1080-1102.

[7] Meng, L., Zhou, X. Simultaneous train rerouting and rescheduling on an *N*-track network: A model reformulation with network-based cumulative flow variables. Transportation Research Part B 67, 208-234.

[8] Pacciarelli, D., 2013. Introduction and classification of existing rescheduling literature. Course lecture in the 5th International Seminar on Rail Operations Modeling and Analysis (RailCopenhagen 2013), Technical University of Denmark, Denmark. http://indico.conferences.dtu.dk/internalPage.py?pageId=8&confId=120.

[9] Pallottino, S., Scutellà, M. G., 1998. Shortest path algorithms in transportation models: classical and innovative aspects. Equilibrium and Advanced Transportation Modelling, (Eds.) Marcotte, P., Nguyen, S., Kluwer Academic Publishers, MA, 245-281.

[10] Zhou, X., Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. Transportation Research Part B 41 (3), 320-341.

[11] Ziliaskopoulos, A., Mahmassani, H. S., 1993. A time-dependent shortest path algorithm for real-time intelligent vehicle/highway systems applications. Transportation Research Record 1408, 94-100.